Google Landmark Recognition 2021

1 Intro

Do you sometimes wonder what it is behind you in that photo, what's that temple or mountain? This technology could predict landmark labels to help you better organize your photo albums, or just to let you be aware of what it is in your photo. This topic refers to Kaggle competition. Google Landmark Recognition should predict landmark labels directly from image pixels. Landmark recognition should not only assign the correct classification, it should be also capable of defining of what landmark is and what not. However, in this project we will try to keep it simpler and train the CNN only to recognize correct classification of image and finding other similar images.

2 Dataset

The dataset used is downloaded from Kaggle and it has around 5 million of images and 81 313 distinct instance labels (our landmarks). This dataset is really big and contains landmark images as well as non-landmark images. In a file, where are image labels assigned correctly to image ids, we added path to each image file, therefore when accessing the images we skip those, which are non-landmark. All images were rescaled by value 255. We needed to use Colab Pro due to big amount of data and our image size is 64. Still we were not able to train the dataset on all data, because we ran into troubles with RAM. So we did it on 44 labels, which has altogether 30 281 images.

Figure 1: Data

	id	landmark_id	path
0	17660ef415d37059	1	./train/1/7/6/17660ef415d37059.jpg
1	92b6290d571448f6	1	./train/9/2/b/92b6290d571448f6.jpg
2	cd41bf948edc0340	1	./train/c/d/4/cd41bf948edc0340.jpg
3	fb09f1e98c6d2f70	1	./train/f/b/0/fb09f1e98c6d2f70.jpg
4	25c9dfc7ea69838d	7	./train/2/5/c/25c9dfc7ea69838d.jpg

3 Methods

In our work we used CNN which was already de- signed – ResNet-50, which consists of 50 deep layers but we did not use any initiated weights and we used Adam as an optimizer. We added callback Early Stopping to stop the training

when the metric is not improving significantly. The activation function used in the output layer is Softmax.

Figure 2: Model

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 2, 2, 2048)        23587712

 flatten (Flatten)           (None, 8192)              0

 dropout (Dropout)           (None, 8192)              0

 dense (Dense)               (None, 44)                360492

=================================================================
Total params: 23948204 (91.36 MB)
Trainable params: 23895084 (91.15 MB)
Non-trainable params: 53120 (207.50 KB)
_________________________________________________________________
```

We also tried preprocessing of images to augment our images in training data, but we did not see any better results (CNN worked really good even without it).

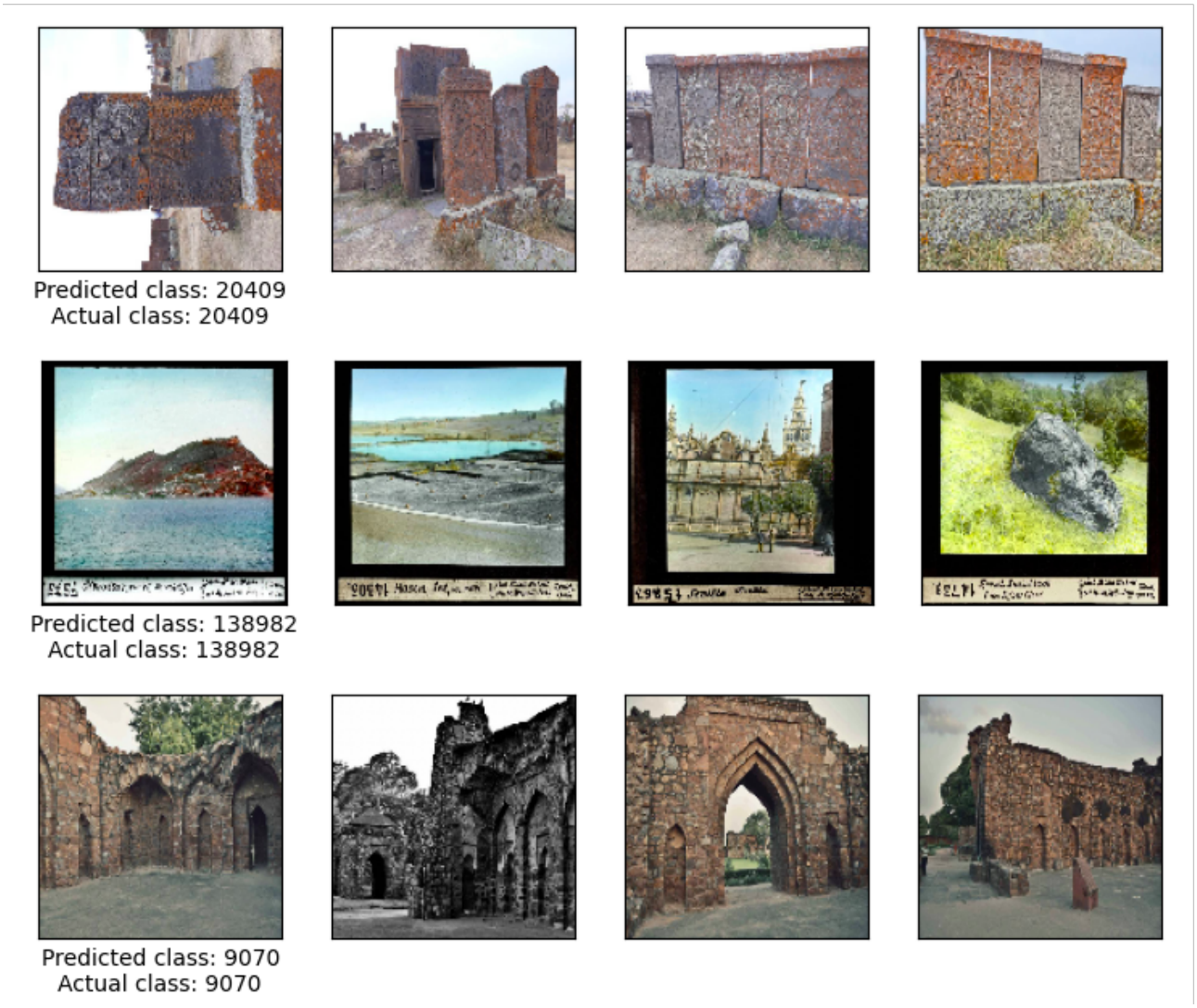
Figure 3: Image augmentation

4 Results

On our little sample of data it works almost perfectly. We used 44 classes but they had around 1000 images per one class.

Below, we can see an example of what we pre-dicted and what was the actual class.

Figure 5: Image results



We also prepared python file for deploying AI to Streamlit, but we were not able to figure out how to load such a big trained model there.

5 Conclusion

From the result we can say that ResNet-50 is really effective. However, it has so good results because we tested it only on the dataset with only those labels, on which it had been trained. It is sad that we did not have the resources to train it on the whole dataset or more labels, because then we would really see what it is capable of. CNNs are powerful, but all data preparation, choice of model and its parameters have to be always taken into careful consideration.

References

[1] Google landmark recognition 2021. online, 2021. [cit. 2018–10–11] https:// www.kaggle.com/ competitions/landmark-recognition-2021/ overview.